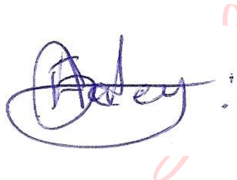
Gradistive Axiomatization for Progressive Semantic Compression in Anisentropic Metric Spaces



Digitally signed by Femi Fadeyi
DN: cn=Femi Fadeyi, o=Virtual Software Institute Inc., ou=Mechanical Cognition Scientist, email=femi@vsinstitute.com, c=US
Date: 2016.07.25 21:01:00 -04'00'
Adobe Acrobat version: 11.0.3

Femi Fadeyi
Mechanical Cognition Scientist
Virtual Software Institute Inc.
femi@vsinstitute.com

Abstract

Gradistive Axiomatization is a form of prioritization and semantic expression which allows encoding arbitrary objects and describing their membership to arbitrary sets in anisentropic metric spaces; as such I believe that Gradistive Axiomatization is a form of mechanical cognition: Progressive Semantic Compression in Anisentropic Metric Spaces. This should be evident in the performance of GA in… compression in anisentropic metric spaces?

1. Introduction

1.1. This hypothesis was developed as part of Project Edison to address the **hard problem of AI**; which I believe is the set-membership / auto-compression problem.

1.2. I formulated this hypothesis based on implication congruence (Implicational Propositional Calculus and Hypothetical Syllogism), which resulted through an attempt to clarify Mutualistic Intent Refinement, Prioritization and Expression. I originally felt that context set theorem would be useful for set membership computation. I eventually discovered AIXI and decided that prioritization is salient / interesting. I had realized that intent refinement, prioritization and expression are somehow relevant to the hard problems of AGI.

2. Observations: Possibly useful truths; Meta-Expectations; Why ask the question;

2.1. Multi-armed bandit problem

2.1.1. I recognized that applying the prioritization problem to algorithms (instead of sentences) is essentially the multi-armed bandit problem. And the multi-armed bandit reformulation is one way to deal with (address) decision problems.

2.1.2. Mitric implications might allow commoditization / normalization / distribution of processing with separation of concerns and atomic validity?

2.2. Semantics and decisions

2.2.1. Semantics let me account for my beliefs about the problem by talking about the consequences of decision methods. So basically, I write out the next sequence of operations that I think would be required based on the result of an execution. Helps me account / keep track of the necessary consequences of some of my beliefs given other beliefs to maintain consistency - Transience / inheritance of consequences and antecedents

2.2.2. This allows me to write conditional branches as implications.

2.2.3. The network doesn't tell me anything about which possible new connections might be useful, however, it does tell me which new connections do not fit well with the existing knowledge about the problem space.

2.2.4. Helps me integrate / account / keep track of new beliefs and observations - Progressive integration of beliefs and observations I can then progressively introduce new algorithms that I discover / invent / construct; and add the things I know to be true about these new algorithms in terms of dependence on other algorithms. This essentially creates classes of algorithms, relatedness.

2.2.5. Helps distribute the actual truths / consequences based on actual observations to resolve open problems - I want to utilize observations to resolve relevant open problems.

3. Question: What is a useful mechanism for Compression in Anisentropic Metric Spaces?

3.1. Conceptually, a set membership problem is a problem where the goal is to decide if any arbitrary candidate object belongs to a particular set. The general set membership problem (the hard problem) asks for a mechanical process to making such decisions for any arbitrary set.

3.1.1. The essence of such problems seems to lie in precisely deciding the equivalence of arbitrary representations of equivalent objects.

3.1.2. A set membership problem is equivalent to the formulation where a machine is given an incomplete partial function definition, and it is tasked to decide if an arbitrary point (x,y) coordinate is on the graph or not. The problem is that there is no effective mechanistic process for selecting which point lies on an arbitrary incomplete graph, at least not in a way that retains any guarantees about our expectations for a valid solution.

3.1.3. The general set membership problem may not be mechanically addressable directly.

3.2. The mechanical thinking problem is a problem where the goal is to prepare a system that can learn in various circumstances and use this knowledge in various other circumstances.

3.2.1. This is equivalent to attempting to extract useful knowledge from previously unknown instances and apply this knowledge in further unknown instances. Equivalently describe a problem given instances?

3.2.2. The "variety" of circumstances will dictate the difficulty of preparing such machines which follow rules precisely, the difficulty lies in writing rules for various instances. The variety of circumstances that machines can face in the real world can be

incredibly high. In fact, for many practical applications, the variety is so high that one struggles to present sufficient coverage of scenarios for the machine to be useful.

3.2.3. To deal with such real-world problems, the current art finds a way to separate a complex problem using randomized examples and identifying the most useful properties, or sets of properties, for separating instances and making such decisions. Such methods, though practically powerful, are non-constructive - The system cannot easily explain its justifications nor reason about justifications and abstractions directly.

3.2.4. The mechanical thinking problem looks ahead to find ways to address scenarios with relatively-low occurrence rates (where randomization techniques may struggle to present exhaustive separable examples) or generalizing knowledge from one domain into another domain (which I want to show is the same problem).

3.3. The key requirements are extracting useful knowledge from various scenarios and generalizing knowledge usefully. These are equivalent to the set membership problem described earlier. As a result, I believe that if one finds a mechanism to effectively deal with set-membership problems in a general way, such a mechanism would be useful for developing thinking machines.

3.3.1. What is a mechanism to effectively tackle set-membership problems?

4. Expectations (What)

4.1. Description of my wants

4.1.1. I want the machine to progressively accumulate new information as beliefs (knowledge) that fit with the intent of my problem statement

4.1.2. I want the machine to judiciously make use of the knowledge that it has in different scenarios. (Separability)

4.1.3. I want the machine to pursue consistency. As long as the machine remains consistent with what I told it to believe, I won't get mad at it for believing something that I didn't properly account for. I will not expect a set of beliefs that is *logically* consistent with my initial set of beliefs to be less useful.

4.1.4. I expect that a machine which does not violate consistency and takes advantage of free exploration, and usefully resolves inconsistencies, would be useful for addressing set membership problems.

4.2. Definition of Anisentropic Metric Space

4.2.1. The metric spaces that I want to work with are sets (which are probably high-dimensional) in conjunction with entropic metrics; where each metric (physical dimension) may be associated with a measure of entropy gradient (time?)

4.3. Stability / Relevance

4.3.1. The system expects that it is able to perform relevant experiments / or at least that:

4.3.1.1. 1. The entropy gradient is a useful measure of relevance between output and input.

4.3.1.2. 2. The limit of the performance of the system will be conditioned on how well its actions / experiments actually result in changes in the environment

4.3.1.2.1. The key here is that the results I am showing you are pertinent to the experiment that you think you're performing semantically

4.3.1.2.2. more important than the environment being somewhat stable, is that I don't manipulate the experimentation process arbitrarily

4.4. Definition of Progressive Semantic Compression

4.4.1. The goal of progressive semantic compression is to identify (and/or create) semantic theories about observed trends; where semantic theories are describable using a mutually-agreed language and possibly through exemplification as well.

4.4.2. PSC Performance

4.4.2.1. Descriptive relevance

4.4.2.2. Descriptive simplicity

4.4.2.3. Predictive accuracy

4.4.2.4. One-shot speed

4.4.2.5. Semantic consistency

4.4.2.5.1. Semantic Settling time (time required to learn what the root words mean)

4.4.3. How exactly are semantics conceptualized here?

4.4.3.1. Phonemes?

4.5. Definition of Progressive Axiomatization (per Theory of Progressive Axiomatization)

4.5.1. When one utilizes prioritizations of theories to control a partially-uncontrolled experimentation process, Progressive Axiomatization asserts that the set of prioritized theories which most contributed to an observation (or statement creation) is conceptually equivalent to the justification (cause) for that resultant observation.

4.5.1.1. Thus, Progressive Axiomatization theorizes that the information about pre-observation prioritized theories is incredibly useful for distinguishing cause-effect pairs; and is especially valuable because it retains semantic bindings while allowing arbitrary levels of abstraction.

4.5.1.2. The capacity for grounded semantic theories but with arbitrary levels of abstraction allow generating complex models that are explainable and thus subject to even further compression.

4.5.1.3. Looking at prioritized theories helps ensure experimentation relevance (did you actually perform that experiment, or equivalently that is the experimentation you did perform)

4.5.1.4. ? Search for critical, non trivial, minimalistic, consequential, valid, relevant edges ?

4.5.2. Progressive Axiomatization suggests that progressive semantic compression in an anisentropic metric space may be achieved using sequential prioritization composition to formulate meta-theories while following quasi-stable periodic phase

transitions between the following Phases of Progressive Axiomatization, starting with some seed set.

4.5.3. Phases of Progressive Axiomatization

4.5.3.1. Phase 1 - RUN EXPERIMENT / COLLECT DATA

4.5.3.2. Phase 2 - Compound Set Creations and Modifications

4.5.3.2.1. If observed (Effects AND Action), then MAYBE(Action->Effects)

4.5.3.2.2. If observed (Effects AND (NOT(Action))), then MAYBE(NOT(Action->Effects))

4.5.3.2.2.1. This is not equivalent to MAYBE(Action->(NOT(Effects)))

4.5.3.2.3. I want a real object or extant theory that exhibits these properties; Or that joins these concepts together

4.5.3.2.3.1. Creativity is really not so much a fabrication process. It's looking for relevant concepts / objects that might join two currently active, but disjoint notions.

4.5.3.2.3.2. Thus for each statement, I want to also include it's effects / intents; The context within which it is / was useful

4.5.3.3. Phase 3 - Compound Antecedent Selection

4.5.3.3.1. Search If Goal = Effects, then Action

4.5.3.3.2. Select some set of theories as your antecedent. The should probably be somehow relevant to the problem at that specific time

4.5.3.4. Phase 4 - Compound Implication Resolution

4.5.3.4.1. Identify the consequents of the selected theories

4.5.3.5. Phase 5 - Compound Consequent Minimization (Prediction)

4.5.3.5.1. Reduce the consequents to the most supported?

4.5.4. ? Another way of thinking of the phases ?

4.5.4.1. Experimentation Relevance (contextual relevance, confirm all premises are met, load the meta-theory union)

4.5.4.2. Experimentation Confirmation (did I actually run the test, active relevance, if the edge was selected)

4.5.4.3. Experimentation Contradictoriness (did I NOT find the hypothesized result? Check if the result is relevant or partially relevant if Mitric)

4.5.4.3.1. Make sure that no descendant contradictions occur also

4.5.4.4. Experimentation Resolution (bayesian-learn from the observation; partial learning if Mitric)

4.5.5. The periodicity of phases may need to be tuned to the specific project, but may need to be well-defined and reliable

4.5.5.1. ? Should reflect available energy?

4.5.6. Progressive Axiomatization is a process of mapping out the network of actions and effects. Progressive Axiomatization theorizes that this method generates anisentropic sequences of quasi-isentropic sets; in conjunction with meta-sets of such sequences which also further allow meta-sequences of such meta-sets. Minimizations on such generated meta-sets and meta-sequences result in possibly-useful theories which retain exemplification and can thus be described / explained semantically, regardless of complexity.

4.5.6.1. Progressive Axiomatization achieves mechanistic prioritization and expression during natural / simultaneous continuous / semi passive experimentation.

4.6. Description of Gradistive Axiomatization

4.6.1. Gradistive Axiomatization hypothesizes separating Phase Management duties from Priority Calculation duties; using a Tester which changes the active phases on a fairly regular schedule and a Candidate which stochastically nominates theories based on the active phase.

4.6.1.1. The basic idea of any Gradistive mechanism is to stochastically sample from all available extant information with the inclusion of newly available information, but with a sampling distribution over multiple epochs.

4.6.2. Gradistive Axiomatization elects to consider a Candidate and a Tester partly to make system design a bit more modular. And I elect to first try a candidate that operates stochastic sampling spread over all time for search and nomination processes. The operation of the Tester seems rather generic / generalizable.

4.7. Axioms of Gradistive Axiomatization

4.7.1. Root Axiom of Gradistive Axiomatization

4.7.1.1. Any implication that is consistent with the following SetOfSemanticAxioms is in the SetOfSemanticAxioms.

4.7.1.2. Root Axiom is presumed true of the environment by Tester.

4.7.1.3. Semantic Axioms are agreements between candidate and tester.

4.7.1.4. Pragmatic Axioms are expectations of the candidate; to deal with finite resources for operation.

4.7.2. Semantic axioms

4.7.2.1. Semantic Axiom of Description

4.7.2.1.1. Tester and Candidate agree to describe objects and relationships using Complex Mitric Implications.

4.7.2.1.1.1. Positive Real Implication means a positive causation of Consequent by Antecedent (Increase A causes Increase C)

4.7.2.1.1.2. Negative Real Implication means a positive causation of Consequent by Antecedent (Increase A caused Decrease C)

4.7.2.1.1.3. Negative Imaginary Implication means reverse causation of Antecedent by Consequent (Change C cause Change A)

4.7.2.1.2. Tester will be responsible for managing the storage, modification, identification, retrieval and basically all details related to edges.

4.7.2.1.3. Tester will be able to introduce new edges. This will be useful for compound statements (sequences) and for the Candidate to operate competing theories simultaneously.

4.7.2.2. Semantic Axiom of Neoritization (Reasoning by introduction of relevant [contextual] new information)

4.7.2.2.1. Tester and Candidate agree that Nomination is the process by which the Candidate directs the experiment.

4.7.2.2.2. After an experiment / observation performed, Candidate can request the creation compound edges to link compound sets of nominated edges to compound sets of result edges. This is how compound edges are created.

4.7.2.2.2.1. ? To simplify this mode of communication, maybe Candidate only dictates how long to continue the linking process ? Intertial computation ?

4.7.2.2.2.2. The alpha of the new compound edge will be observed entropy gradient. ?The strength will start off at zero?.

4.7.2.2.2.3. ?? Multiple compound edges may be created for different compound echelons of nominees and results

4.7.2.2.2.3.1. First 10 nominated imply first result; First result implies first 10 result; etc?

4.7.2.2.2.4. ((A>+B),(C>-D))>-((W>-X),(Y>+Z))

4.7.2.3. Semantic Axiom of Neodiction

4.7.2.3.1. Tester and Candidate agree that nomination (prioritization) and expression are the same. Thus, Tester is free to use the priorities of the Candidate as semantic expressions on behalf of the Candidate.

4.7.2.3.2. Some “words” are arbitrarily defined?

4.7.2.4. Semantic Axiom of Oblivioritization (Forgetting):

4.7.2.4.1. Tester and Candidate agree that Tester can throw away things Candidate has deemed low priority, If resources are low, to create a kind of Resource-Bounded reasoning process.

4.7.2.4.2. This is like the NOT version of Neoritization by allowing Candidate to assert an inverse preference / when bound by resources.

4.7.2.5. Semantic Axiom of Confidence (Including non-certainty [confusion], WITH)

4.7.2.5.1. Candidate and Tester agree on a common method to express a measure of certainty and what certainty means.

4.7.2.5.2. Candidate can modify Strength of an edge

4.7.2.5.2.1. This is basically just an implication, but Tester must summarize duplicates.

4.7.2.5.2.2. This calculation is a responsibility of the Tester. But Candidate must be agree on how that information is being used.

4.7.2.6. Semantic Axiom of Contradiction (Including non-false [true], NOT)

4.7.2.6.1. Candidate and Tester agree on a common method to express falsehood and what falsehood means.

4.7.2.6.2. Candidate can modify Truth of an antecedent

4.7.2.6.2.1. This is why nodes are required - to sum delta truth.

4.7.2.6.2.2. This calculation is a responsibility of the Tester. But Candidate must be agree on how that information is being used.

4.7.2.7. Semantic Axiom of Freedom

4.7.2.7.1. Tester and Candidate agree that given an active edge, Candidate is free to calculate whatever nomination value it wants for the adjacent edges; using whatever information is available in the active edge and (partially visible portion of) adjacent edges

4.7.2.7.1.1. The specific mechanism used by a Candidate is part of the experimental hypothesis; it is not a part of this axiom.

4.7.2.8. Semantic Axiom of Pragmatism (Resource-Awareness)

4.7.2.8.1. Candidate understands that Tester will inertially drive the focus of the Candidate; roughly splitting time between certain phases. Thus nomination values may be skewed, or even ignored temporarily, per the Pragmatic Axioms that follow.

4.7.2.8.2. This is what allows “semantic” resource distribution because Tester has an expectation and can describe the kind of behaviour a Candidate will perform if Tester allocates more time in a corresponding phase.

4.7.2.8.2.1. Axiom of separation? Epochs?

4.7.2.8.2.2. Phase1 - Compound Antecedent Selection

4.7.2.8.2.3. Phase 2 - Compound Implication Resolution

4.7.2.8.2.4. Phase 3 - Compound Consequent Minimization (Prediction)

4.7.2.8.2.5. Phase 4 - RUN EXPERIMENT / RESHUFFLE

4.7.2.8.2.6. Phase 5 - Compound Set Creations and Modifications

4.7.2.8.3. How will Tester know what Candidate is thinking?

4.7.2.8.3.1. Neodiction...

4.7.2.8.4. What will Tester base phase changes on?

4.7.2.8.4.1. ? Inertia - Candidate’s own goals

4.7.2.8.4.2. ? Distributed / Stochastic

4.7.2.8.4.3. ? Heuristically-interesting extraction (non-trivial)

4.7.2.8.4.4. ? Cross-contextual (possibly usefully pertinent, but previously logically disjoint)

4.7.2.8.4.5. ? Contextually bound (relevant, possibly useful)

4.7.2.8.4.6. ? Separable (semantic)

4.7.2.8.4.7. ? Minimized (non-trivial, filtering, forgetting)

4.7.2.8.4.8. ? Danger

4.7.2.8.4.9. ? Semantic Expression (Neodiction)
4.7.2.8.4.10. ? Emotions?
4.7.2.8.4.11. ? H
4.7.2.8.4.12. ?...
4.7.2.8.4.12.1. THOUGHT. A gradistive mechanism will allow the system to usefully prioritize across multiple problems.
4.7.2.8.4.12.2. Contextually- inertial attention
4.7.2.8.4.12.3. Gradistive = contextually sensitive inertia
4.7.2.8.4.12.4. Focus will semantically be handed to
4.7.2.8.4.12.5. Errors
4.7.2.8.4.12.6. Contradictions
4.7.2.8.4.12.7. New information? Instructions?
4.7.2.8.4.12.8. Random?
4.7.2.8.4.12.9. Explicit input data
4.7.2.8.4.12.10. Guessing + checking

4.7.3. Pragmatic Axioms; Operational Limitations on Prioritization AND Expression

4.7.3.1. ?Pragmatic Axiom of Experimentation?

4.7.3.1.1. Take theory, remove small piece, test
4.7.3.1.2. Take theory, add small piece, test
4.7.3.1.3. ...

4.7.3.2. Pragmatic Axiom of Pleniritization

4.7.3.2.1. (total amount of support)
4.7.3.2.2. Tester expects candidate to express all relevant, true support and to prioritize its time looking for greater quantity of support.

4.7.3.3. Pragmatic Axiom of Miniritization

4.7.3.3.1. (non-repetitive support): Avoid expressing sub-theories of more explanative, relevant meta-theories and to prioritize its time accordingly, looking for different support.
4.7.3.3.2. Non-superfluous? Mellifluous?
4.7.3.3.3. Non-repetitive

4.7.3.4. Pragmatic Axiom of Falsifiability

4.7.3.4.1. (non-triviality possibly false, but not known)
4.7.3.4.2. Tester expects candidate to not waste time on vacuous, inconsequential, observations / truths. Tester should be driven by intent (instantiated prioritizations). Candidate should try to express (extrovert) / compress (introvert) environment.

4.8. Definition of Semantic Network

4.8.1. A semantic network is a set containing cause-effect pairs (implications) that are represented as complex-mitric-valued edges

4.8.2. Complex mitric value

4.8.2.1. A mitric number is any real number contained in the interval [-1,1]

4.8.2.2. A complex mitric number is a number that can be expressed in the form k+α***i***, where k and α are mitric numbers and ***i*** is the imaginary unit, that satisfies the equation $i^2 = -1$.

4.8.2.3. Pure induction (THEN, Implication) is a Positive Real Mitric

4.8.2.3.1. C=k+α***i***; where k>0 and α=0.

4.8.2.4. Pure exception (THEN NOT, Disjunction) is a Negative Real Mitric

4.8.2.4.1. C=k+α***i***; where k<0 and α=0.

4.8.2.5. Pure association (AND, Conjunction) is an Imaginary Mitric

4.8.2.5.1. C=k+α***i***; where k=0 and α≠0;

4.8.2.5.2. α (Implication direction / parentheses) naturally aligns with increasing entropy; Indicates natural / expected progression of experimentation direction; Theory -> Implies -> Observation; Set Definition -> Implies -> Set

4.8.2.5.2.1. This convention seems useful because antecedent can predict many Consequents. But given Consequent, it is difficult to identify Antecedent. Therefore, knowledge about the state of Antecedent provides more information about the state of the world. Therefore, Antecedent knowledge indicates less entropy compared to Consequent.

4.8.2.5.2.2. Every set is a combination of defining Axioms and combination of known elements.

4.8.2.5.2.3. Set ->Implies-> Element seems incorrect

4.8.2.5.2.4. (c→a→t)→(a→n→i→m→a→l)

4.8.2.5.2.5. Since the mitric connective is complex, it can encode this directly as

4.8.2.5.2.6. c←a←t→a←n←i←m←a←l

4.8.2.5.2.7. C[-1]a[-1]t[+1]a[-1]n[-1]i[-1]m[-1]a[-1]l

4.8.2.5.2.8. Post hoc ergo propter hoc; fallacy meaning "after this, therefore because of this"

4.8.2.5.3. Note, an implication that partly defines a set may imply additional expectations which do not belong to the “set”

4.8.2.5.4. Parenthesization is fundamentally necessary for expressing certain logical structures.

4.8.2.5.5.

4.8.2.6. An edge may be thought of as conditional branching in a computer program; where each node is the result of (or trigger for) an algorithm that can be executed by a turing machine.

4.8.3. During evaluation, any statement implied by a TRUE STATEMENT is semantically true (modus ponens). Anything which implies a FALSE STATEMENT is implicitly false (modus tollens).

4.8.3.1. A positive implication means that if the antecedent node evaluates to true (or increases in value) for a particular input, then the consequent is expected to be true (or increase in value) as well. Conversely, if the antecedent is false, no statement is made about the consequent. A negative implication is the inverse of the positive.

4.8.4. ? Anything matching a Tautological structure is implicitly true?

4.8.5. Modification of the network is non-trivial and shouldn't be done arbitrarily

4.8.5.1. Resolving the inconsistencies in policies is non-trivial AND non-trivially changes the knowledge encoded by the network

4.8.5.2. Finding contradictions or integrating new information IS useful!!

4.8.5.2.1. Resolution of congruence is decision making.

5. Hypothesis (Why)

5.1. I want to make an observation that if false, would invalidate Gradistive Axiomatization

5.1.1. If one performs a controlled experiment, but then removes one key element of the GA requirements, the difference in [progressive semantic compression performance] should be clear.

5.1.1.1. The value of GA derives directly from how Progressive Axiomatization achieves progressive semantic compression

5.1.1.2. "The capacity for grounded semantic theories but with arbitrary levels of abstraction allow generating complex models that are explainable and thus subject to even further compression."

5.1.1.3. Progressive Axiomatization recommends utilizing phases to separate prioritization epochs and stochastic shuffling to select priorities.

5.1.1.4. "The goal of progressive semantic compression is to identify (and/or create) semantic theories about observed trends; where semantic theories are describable using a mutually-agreed language and possibly through exemplification as well."

5.1.2. Null hypothesis - I won't be able to tell the difference

5.2. I'm confident GA will show a significant difference because

5.2.1. GA non-trivially conditions the result on an experiment on the environment; in a way that is separable on the biasing premise(s) (expectations). If environment is compressible (multi-variate normal distribution); the biased experimental sampling with axiomatization non-trivially records the most useful (resource-aware) semantic (constructive) interpretations of observations and the environment. This is useful when one is not sure which experiment is being performed at a particular time; or equivalently if one is interested in a more continuous form of experimentation

5.2.2. GA, or more specifically the candidate, doesn't directly look at or care what the environment is doing in any particular state. It instead looks for which of how well various theories perform in describing the environment's behavior.

5.2.2.1. The candidate believes that the choice of which experiment to perform is not trivial. Instead of attempting to test experiments monolithically, the candidate organizes the sub-experiments it finds interesting

5.2.2.1.1. This allows the candidate to possibly test semantic combinations of hypothesis (meta-theory), not just basic hypotheses

5.2.2.2. The candidate requests an external, but consistent mechanism to perform the experiment(s) and return results. The candidate assumes that its order of experiments was taken into account when performing the experiment

5.2.2.2.1. This allows the experimentation system to perform as many micro-experiments as it has resources for, without bothering about the implications / purpose of the micro-experiments.

5.2.2.3. The candidate now looks for which combinations of hypotheses (meta-theories) are most explanative, simplest and most relevant.

5.2.2.3.1. The power of this mechanism is the creation of new meta-theories (of arbitrary complexity) and the introduction of new observational hypotheses while preserving the original core theories and avoiding the selection fallacy (or at least guarding against the selection fallacy, but selecting the most viable theories limited only by processing capacity)

5.2.2.3.1.1. ? And can also learn (and use) competing theories independently?

5.2.3. GA structures the interior of the system as a semantically grounded (separable), metric space. That can create and progressively compress a model of itself in conjunction with the environment.

5.2.3.1. Heuristically-interesting extraction (non-trivial)

5.2.3.2. Cross-contextual (possibly usefully pertinent, but previously logically disjoint)

5.2.3.3. Contextually bound (relevant, possibly useful)

5.2.3.4. Separable (semantic)

5.2.3.5. Minimized (non-trivial, filtering, forgetting)

5.2.3.6. Prioritization sequencing (actions, goals)

5.2.4. GA produces what we would describe as a visualization of solution / problem, then modifies this rationally to produce alternate theories

5.2.4.1. Just take one object and another object and say, sure, that should work. That's the easy part.. that's "Visualizing a solution"

5.2.4.2. Now, the hard part is trying to describe how they are related in words.

5.2.4.3. Also, deciding when something is salient enough (has occurred again, and is possibly useful) to deserve a name - and subsequent analysis / definition

5.2.4.3.1. Probably by just observing a nomination of something linking them when quieting some of the noise

5.2.4.3.2. Or some logical structure?

5.2.4.4. This is how one formulates a "target solution" an idea

5.2.4.5. And then one can use this to filter / attract more descriptive statements

5.3. For this experiment, I will calculate nominations as follows:

5.3.1. Experimentation Relevance (contextual relevance, confirm all premises are met, loaded the meta-theory union completely)

5.3.1.1. How do I verify experimental setup matches Gradistive Axiomatization?

5.3.1.2. Check the Axioms?

5.3.2. Experimentation Confirmation (did I actually run the test, active relevance, if the edge was selected)

5.3.2.1. ?..

5.3.3. Experimentation Contradictoriness (did I NOT find the hypothesized result? Check if the result is relevant or partially relevant if Mitric)

5.3.3.1. Evaluate the Results

5.3.3.2. What does it mean to do well

5.3.4. Experimentation Resolution (bayesian-learn from the observation; partial learning if Mitric)

5.3.4.1. Update my Beliefs

5.3.5. Do this all atomically (with periodic phases)

6. Experimental Setup (How)

6.1. Edison.JS System Outline: https://docs.google.com/document/d/1bLiPY5aGz82iCumI0E2fKOwOg09UBIEtCLebpdmoHu8/edit

6.1.1. How much computation resource do I need?

6.2. I want to investigate how the different properties of GA correlate with the following properties of progressive semantic compression performance

6.2.1. Descriptive relevance / consistency: Not based on the "correct" answer, but based on the system's own expressed goal.

6.2.2. Descriptive simplicity: How long are descriptions?

6.2.3. Descriptive efficiency: How much non-overlap are in the description components?

6.2.4. Predictive accuracy: How well does the system actually perform in making predictions?

6.2.5. ? One-shot speed? ? How many repeated exposure to the same sequence does it take before the system recognizes this sequence

6.2.6. ? Semantic consistency ? Semantic Settling time (time required to learn what the root words mean / start using the same words consistently?)

6.3. Experimental constraints

6.3.1. The ideal test should be relevant to the capabilities of the system and be semi-stable

6.3.1.1. Don't test a 6 year old at the same level as a 15 year old. Don't test a human on binary data Don't expect a human brain to usefully process trends over gigaseconds; or femtoseconds

6.3.2. The ideal test should evaluate the system's consistency in its descriptions

6.3.2.1. The system tells me what it believes will happen AND why; then based on this statement, I show a relevant consequence of that belief in my world

6.3.2.2. Then I expect the system to describe what it think happened and present a new theory (which may be a clarification of an existing theory with new words)

6.4. Crawl-versation Experiment

6.4.1. Conversations are helpful for training for conversations; thus I propose a “conversation” where responses are served by a web crawler (wiki / google / etc); ergo “crawl-versation”

6.4.2. Semantic basis

6.4.2.1. Letters seem helpful because to test the system, I don’t need a huge semantic dictionary; but this has the consequence that letters have basically zero semantic value and I can expect the system to struggle to build up its internal semantic dictionary. Though, my semantic dictionary (training resource) would just be the size of the alphabet.

6.4.2.2. Complete words will require a much larger semantic dictionary than letters, but I’m not worried about the technical challenge - I can find techniques to manage this. I am so concerned about semantic effectiveness and I want my experimental setup to show this bias. This semantic dictionary would probably be the size of a typical vocabulary; maybe 5,000 - 30,000 words.

6.4.2.3. Morphemic / Lexemic dictionary may add even more semantic richness that might not be available in a standard dictionary, while only having a semantic dictionary size of around 2,000-5,000 words.

6.4.2.4. I want to structure part of my test language as follows:

6.4.2.4.1. The primary elements are morphemes; which can be combined into sets which I can then provide better definitions for.. If there is a near word match

6.4.2.4.2. http://alexandstein.com/main/morphemes

6.4.2.4.3. http://www.cognatarium.com/cognatarium/

6.4.2.4.4. I’ll just try inverse-parsing using a dictionary relevance search

6.4.2.4.5. I'm pretty confident I can hack one together by just using a dictionary relevance search. Example: a basic relevance search for [epi, tome] will return epitome probably highest, then I will define [epi + tome] = [epitome]. But I'm curious if there is a more elegant solution for this problem, that actually evaluates the

6.4.2.4.6. Many words are not composite in this way, but I think having this mechanism of word synthesis / lookup will be central to the testing process anyway.

6.4.2.4.7. But it does mean that I will need to expose the system to such composite words whenever it sees them?

6.4.2.4.7.1. Not necessarily.

6.4.2.4.7.2. It’s just part of the total information available when the system asks for clarification about a word

6.4.2.4.7.3. Letters are so irrelevant that the system doesn’t really need to learn them

6.4.2.4.7.4. I need to provide handles anyway, so the system can make connections between words and ask for definitions directly

6.4.2.4.7.5. The system will kind of need to understand that words are composite on their defining concepts and even separable on context

6.4.2.4.8. But I do want to separate clarification from conversation.

6.4.3. Unfortunately, this is still a partial solution - it will learn to talk, and read; but not really listen or converse.

6.5. GameWalker Experiment

6.5.1. System expectations

6.5.1.1. The system is expected to understand how action vectors (pathlets) work and how they relate to navigating its environment

6.5.1.2. The system is expected to understand what people look like in its environment

6.5.1.3. The system is expected to identify the person / people closest to its goal goal vector

6.5.1.4. The system is expected to navigate to the person nearest its goal vector / destination

6.5.1.5. The system is expected to recognize if it is on a path that won't work and stop trying that path

6.5.1.5.1. Impeded motion

6.5.1.5.2. Or no working forward paths from current location

6.5.1.6. The system is expected to remember the layout of its environment so it doesn't try paths it already should know don't work

6.5.1.6.1. It might also want to remember specific locations / paths that do work well.

6.5.1.7. The system is expected to indicate if it is unable to find a workable path

6.5.2. The system publishes hypotheses about where in video (pathlets) it thinks the target is and it can combine pathlets into meta-pathlets, as it attempts to figure out which meta-pathlets to check which will most provide information about target.

6.5.3. Setup Considerations

6.5.3.1. I evaluated the possibility of

6.5.4. Not sure which game yet

6.5.4.1. https://bitbucket.org/osrf/gazebojs; http://gazebosim.org/download

6.5.4.2. http://wiki.roboteducation.org/Calico_Simulator

6.5.4.3. https://en.wikipedia.org/wiki/Webots

6.5.4.4. https://github.com/rdiankov/openrave

6.5.4.5. http://simbad.sourceforge.net/?cm_mc_uid=72611389141414694662254&cm_mc_sid_50200000=1469469548

6.5.4.6. https://github.com/jonklein/breve.js

6.5.4.7. https://en.wikipedia.org/wiki/Microsoft_Robotics_Developer_Studio

6.5.4.8. https://en.wikipedia.org/wiki/Player_Project

6.5.4.9. https://phidgets.wordpress.com/2014/05/05/getting-started-with-phidgets-on-the-raspberry-pi/

6.5.4.10.

6.5.5. Requirements

6.5.5.1. I really just need a mechanism that provides stereoscopic video of a realistic environment

6.5.5.1.1. System's position

6.5.5.1.2. System controls

6.5.5.1.3. Visual data

6.5.5.1.4. Obstacles

6.5.5.1.5. Target / goal position

6.5.5.2. And a mechanism for the system to try walking and learning to avoid / describe obstacles

6.5.5.3. I'll know that the system is performing well if it successfully

6.5.5.3.1. Goes to target destination given heading vector

6.5.5.3.1.1. GPS location of current and target might be used to compute a heading and distance vector

6.5.5.3.1.2. When vector length drops below a certain value - goal is reached

6.5.5.3.2. Understands when it has run into an obstacle

6.5.5.3.3. Avoids visual obstacles

6.5.5.3.4. Will take the longer way around if necessary to avoid an obstacle

6.5.5.3.5. Remembers shortcuts

6.5.5.4. Might be able to test with a simple / real robot

6.5.6. Will probably need to give some feedback so the system prefers certain outcomes

6.5.7. Representation synthesis

6.5.7.1. Integrate information from GPS orientation subsystem; GPS path options subsystem; binocular vision subsystem and the Trainer RYG subsystem

6.5.8. Visual processing ties into natural language processing as follows

6.5.8.1. One can consider each frame of video a sentence (a clearly, separable unit with some sequential tie to preceding and anteceding statements)

6.5.8.2. Each pixel is a letter; has no direct / useful meaning

6.5.8.2.1. This isn't completely true in the case of pixels, since a single pixel might distinguish certain classes of objects

6.5.8.2.2. But one could perhaps make a similarly weak case for using single letters to identify classes of concepts..

6.5.8.2.2.1. The lack of the letter X is a useful way to separate ideas about xylophone from ideas not about xylophones

6.5.8.2.2.2. Of course, however, I can construct an infinite many sentences sentences about xylophones without ever using the letter X. And an equivalent many sentences with the letter X that are also not about xylophones; *hoc exemplum hoc*.

6.5.8.3. Contained within this sentence, however are meaningful and separable concepts but described awkwardly. Letters are shuffled and rearranged, and spaces may not be completely preserved.

6.5.9. What is a description?

6.5.9.1. Well, it's important to first understand the semantic basis for the system

6.5.9.2. There are ground / root connections which tie to particular outputs and inputs

6.5.9.3. The goal is to wire the response such that a particular trend in the input will be predicted to the most likely sequence of inputs

6.5.9.4. Thus if you have such a prediction, you can use it to tie to specific outputs that maximize a particular goal

6.5.9.5. E.g. if ((A-,B+,C+,D-,F+)then(A-,B-,C+,D-,F-)then(A-,B-,C+,D-,F+))then((A-,B-,C-,D-,F+)then(A-,B+,C+,D+,F-)) can be established as a meta-hypothesis

6.5.10.

6.5.11. Descriptive relevance / consistency: Not based on the “correct” answer, but based on the system’s own expressed goal.

6.5.11.1. E.g. the system has an interest in checking a particular set of hypothesis

6.5.11.2. I want to see how it’s reaction to new information

6.5.11.3. Indicates that it understands (or is at least aware of) what it was trying to do

6.5.11.4. --

6.5.11.5.

6.5.12. Descriptive simplicity: How long are descriptions?

6.5.13. Descriptive efficiency: How much non-overlap are in the description components?

6.5.14. Predictive accuracy: How well does the system actually perform in making predictions?

6.5.15. ? One-shot speed? ? How many repeated exposure to the same sequence does it take before the system recognizes this sequence

6.5.16. ? Semantic consistency ? Semantic Settling time (time required to learn what the root words mean / start using the same words consistently?)

6.6. Let’s say the output from the system is used by the environment to generate the next input

6.7. TEST PROTOCOL:

6.7.1. Test Bench connects with seed data

6.7.1.1. Each edge represents an element of a document; may be a word sequence literal or a parenthetical connective (associating groups of words)

6.7.1.1.1. It is important to test the ability of the system to work with separable structures, parentheses help

6.7.1.1.1.1.

6.7.1.1.2. Page (NEW_PAGE)

6.7.1.1.3. Paragraph (NEW_LINE (\n))

6.7.1.1.4. Sentence (PERIOD (.), COMMA (,), SEMICOLON (;))

6.7.1.1.5. Fragment (PARENTHESES ((,),],[,<,>))

6.7.1.1.5.1. I might also want to consider lists

6.7.1.1.6. 3-word sequence; or Rand (2,5)

6.7.1.1.7. Word (SPACE ( ))

6.7.1.2. Each edge has the following attributes

6.7.1.2.1. Antecedent Text

6.7.1.2.2. Consequent Text

6.7.1.2.2.1. May be the same as antecedent for a self-edge

6.7.1.2.3. Real Strength - A floating-point number between -1 and +1, inclusive. Indicates that a word is implied by another word; e.g.

{(0.8+0i) DOG -> ANIMAL}, {(0.1+0.0i) paragraph_1 -> paragraph_2}, {(0.05+0.0i) paragraph_1 -> paragraph_3}

6.7.1.2.4. Imaginary Strength - A floating-point number between 0 and +1, inclusive. Indicates the belongingness of an element to a set (like a fragment, or page); e.g. {(0+1.0i) page_1 -> paragraph_1, page_1 -> paragraph_2, page_1 -> paragraph_3}

6.7.2. Raw Performance Test Bench (Every operation +1 CountAllTicks) - Tests algorithmic efficiency in storing information to make it easily searchable for itself

- 6.7.2.1. READ_NEXT_INPUT (+1 CountAllInputs, +1 CountIfNewInputs)
- 6.7.2.2. CONTEXT_STACK_REPRIORITIZE
- 6.7.2.3. FOCUS_DOWN_CONTEXT_STACK
- 6.7.2.4. FOCUS_UP_CONTEXT_STACK
- 6.7.2.5. FOCUS_FIRST_CHILD
- 6.7.2.6. FOCUS_FIRST_PARENT
- 6.7.2.7. FOCUS_NEXT_SIBLING
- 6.7.2.8. FOCUS_PREV_SIBLING
- 6.7.2.9. REAL_STRENGTH_PLUS
- 6.7.2.10. REAL_STRENGTH_MINUS
- 6.7.2.11. IMAGINARY_STRENGTH_PLUS
- 6.7.2.12. IMAGINARY_STRENGTH_MINUS
- 6.7.2.13. NOMINATION_PLUS
- 6.7.2.14. NOMINATION_MINUS
- 6.7.2.15. INSERT_INPUT_AFTER
- 6.7.2.16. INSERT_INPUT_BEFORE
- 6.7.2.17. INSERT_INPUT_CHECK_HERE
- 6.7.2.18. INSERT_INPUT_HERE (+1 CountIfHits)

6.7.3. Raw Performance = CountIfHits/CountAllTicks * CountIfNewInputs/CountAllInputs

6.7.4. Semantic Performance Test Bench

- 6.7.4.1. READ_BACKWARD
- 6.7.4.2. READ_FORWARD
- 6.7.4.3. BUFFER_CLEAR
- 6.7.4.4. BUFFER_APPEND
- 6.7.4.5. GO_WIKI_BUFFER
- 6.7.4.6. GO_THESAURUS_BUFFER
- 6.7.4.7. GO_DICTIONARY_BUFFER
- 6.7.4.8. GO_GOOGLE_BUFFER
- 6.7.4.9. GO_NEWEST_MESSAGE
- 6.7.4.10. GO_PREV_MESSAGE
- 6.7.4.11. GO_NEXT_MESSAGE
- 6.7.4.12. GO_LINK
- 6.7.4.13. CERTAINTY_PLUS
- 6.7.4.14. CERTAINTY_MINUS
- 6.7.4.15. CONFUSED_PLUS
- 6.7.4.16. CONFUSED_MINUS
- 6.7.4.17. --
- 6.7.4.18. NEW_MESSAGE_AVAILABLE

6.7.4.19. FEEDBACK_PLUS (
6.7.4.20. FEEDBACK_MINUS
6.7.4.21. ENERGY_PLUS (Automatic, for BUFFER_APPEND, PLUS/MINUS actions)
6.7.4.22. ENERGY_MINUS (Automatic, for all actions)
6.7.5. Semantic Performance = ?
6.7.5.1. Think of entropy
6.7.5.2. And the goal of the machine is to prioritize actions which help it to gain a better understanding of its environment
6.7.5.3. It is, in essence, an extension of the raw performance.
6.7.5.4. But now, I care about what you choose to do and how you feel about it
6.7.5.5. I think semantic evaluation mandates approximation
6.7.5.6. The right answers are embedded within a answer space and your correctness is measurable
6.7.5.7. I want a system which
6.7.5.7.1. minimizes the context set
6.7.5.7.2. identifies inconsistencies
6.7.5.7.3. Isn't overly confident when it is incorrect
6.7.5.7.4. Is confident when it is correct
6.7.5.7.5. Asks good questions
6.7.6. Once energy runs out
6.7.6.1. System is evaluated for performance,
6.7.6.2. I expect that the system will basically start with the context (its goal) and then try to find interesting relevant information in the document (s)
6.7.6.2.1.
6.7.6.3. Candidate System is reorganized according to its self-modification specification
6.7.6.4. If testing many systems, energy may be distributed based on relative performance.
6.7.7. Since the system's previous question (or answer) is in the modified context set, it might be a good idea to provide it a semantic answer to its question.
6.7.7.1. Maybe this should be done along the way?

6.8. EXPERIMENTAL HYPOTHESIS. A mechanical process following this process will provide relevant summaries of text and can be set up to express unclarity.

6.9. Start with an energy budget
6.10. Distribute energy budget across a set of mitric implications each with a contextual weight
6.10.1. With may feature parenthetic boundaries
6.11. Walk around prioritizing
6.11.1. I = Historically Valid, Equivalent rooted strength of nominee ancestors x nominee's strength
6.11.2. P = Contextually Recent, Equivalent age-corrected rooted strength of nomination
6.11.3. D = Actively Relevant, Equivalent variant-intersectoriness
6.11.3.1. How many children from different contexts are dependent on this parent node?

6.11.4. H = Anticipatorily False, Equivalent auto-contradictoriness
6.11.4.1. How much does this parent contradict itself
6.12. Flag the nominator that nominated each context edge using a context binding / instantiation / shortcut
6.13. Flag Semantic Answers: If you find a contradiction nominate FALSE, if you find a nexus nominate SELF.
6.14. Remove non-critical resultants from context set
6.14.1. If an edge was not originally in context set and context has parent which explains the edge, then the edge is non-critical (though possibly not trivial either).
6.15. Report progress as size of context set
6.15.1. Or rather the speed of reduction in size?
6.16. Pay energy bill
6.17. Once energy runs out, or if interrupted
6.17.1. Add any new information available (but try to retain some of the previous resultant context)
6.17.2. HOW DO I IDENTIFY KNOWLEDGE GAP?
6.17.2.1. Little minimization progress / energy expenditure?
6.17.2.2. I should probably meta-prioritize these areas
6.17.3. HOW DO I VERIFY MINIMIZATION?
6.17.3.1. Look forward to make sure minimization indeed captures the seed set
6.17.3.2. Identify which ones do so
6.17.3.3. Re-run these to see if they remain minimal
6.17.3.4. I should probably meta-prioritize these areas
6.17.4. HOW DO I RESOLVE CONTRADICTIONS?
6.17.4.1. Look for a specific kind of input data?
6.17.4.2. ASK A QUESTION!!
6.17.4.3. Focus attention on something?
6.17.4.4. Find out who to blame in the seed context set and describe it
6.17.4.5. What's the difference between a cat and a penguin?
6.17.4.6. What counter-example would minimize the error space here?
6.17.5. I SHOULD PROBABLY PROPAGATE THIS KNOWLEDGE
6.17.5.1. Well, the object causes a contradiction.. I should probably trace back the effect this has on parents.. Flag all the supporters as "uhm, we might have a problem here"
6.17.5.2. If someone correctly predicted it was wrong, you might say "uhm, I think this guy was on to something" and probably the same about its consequents..
6.17.5.3. This process is semantic (as in respects relevance (inertia), probably useful (heuristic), but is not logically supported)
6.17.5.4. This process is meta- because it starts talking not just about the objects of discourse (edges), but the observed sequences of these due to the systems own prioritization process and context binding (shortcutting)
6.17.6. You can update strengths gradistively to test what-if hypotheses
6.17.7. If you were performing a what-if hypothesis, don't forget to update the strength terms, especially for the viktors of this what-if round)
6.18. AGI = gradistive-what-if(ixI,pxP,dxD,hxH)

6.18.1. Symbols / mathematical concepts / representations ARE ulterior prirortizations (sequences) / descriptive nexi embodied in a finitive network of finite resources RESPECTFUL OF minimizations and free contradictions.
6.18.2. It is possible to select an arbitrary real number by simply giving it a name and a description, like 1, or PI.
6.18.3. Prioritized allocation of semantic (context-rooted) symbols to concepts is intelligence
6.18.4. Usually it’s best to prioritize hypotheses which are minimalistic and general.
6.18.5. Usually it’s best to identify and try to resolve contradictions

7. Experimental Results (Data)

7.1. I need an example of a valid decision problem scenario where the addition of new semantic information subsumes / resolves the indeterminacy.
7.2. Pending…

8. Thoughts (Meta-Implications, Philosophy, Discussion, Conclusions)

8.1. Semantic Computation Meta-Definition

8.1.1. Such an arrangement (per the Gradistive Axioms) between a Candidate and a Tester implements a Semantic Computer, with the Candidate as the Algorithm and the Tester as the Processor / Interpreter.

8.2. Auto-Compression Meta-Hypothesis

8.2.1. (not part of this project directly, but I feel it is philosophically)
8.2.2. Progressive Semantic Compression in anisentropic metric Spaces is the same thing as cognition.

8.3. It is interesting AND important AND illustrative to note that **modification does NOT take into account the success or failure of ANY particular algorithm**, it is only and fully engaged with the prioritization and structure of policies while maintaining certain guarantees.

9. Open Questions (Further Work)

9.1. What properties can I remove or relax from Gradistive Axiomatization, but still retain capacity for Semantic Computation (Progressive Semantic Compression in anisentropic metric Spaces)?
9.2. How do I measure effectiveness of GA for PCAMSing?
9.3. How do changes in the properties of GA modify it's effectiveness?
9.4. Is there a way to check if pcamping is general enough to define semantic computation? Or is there no way to know this? Should pcamping be axiomatized, used as the accepted, arbitrary, definition of Semantic Computing (Well Defined (contextually bound) Mechanical (algorithmic) Abstract Reasoning)
9.5. ? It may also be interesting to consider “rests” over a longer time frame
9.5.1. Rest allows the brain to reduce the noise from all the candidates there were nominated, but that ended up not matching well. But flagging the good candidates should be done live (inner process).

9.6. It appears that the brain physically allocates different regions of the brain to handle necessities, beliefs, chores and ideas. How are these constructs represented in Gradistive Axiomatization?

9.6.1. It seems that the introduction of changing phases itself doesn't handle the specific need to consider nominations from different epochs

9.6.2.

10. Random + Notes

10.1. Note: We are incredibly smart in our ability to grasp new info. But we are incredibly slow in doing so, without representative forms / mutual structures / constructive basis / bias / useful semantics / relevance.

10.2. http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

10.3. http://mldata.org/

10.4. http://archive.ics.uci.edu/ml/

10.5. http://homepages.inf.ed.ac.uk/rbf/IAPR/researchers/MLPAGES/mldat.htm

10.6. http://www.cs.toronto.edu/~delve/data/datasets.html

10.7. http://deeplearning.net/datasets/

10.8. Text corpora - http://disi.unitn.it/moschitti/corpora.htm

10.9. http://disi.unitn.it/moschitti/corpora.htm

10.10.